

PLACID : une Plateforme pour Coopérer en Conception Distribuée

Alain-Jérôme Fougères, Max Recchione, Yaël Gomez

¹Laboratoire M3M - UTBM

Rue du Château Sévenans

90010 BELFORT - France

alain-jerome.fougeres@utbm.fr

Résumé— Nous relatons dans cet article la conception d’une plate-forme multi-agents adaptée à la conception innovante et distribuée de systèmes mécaniques et supportant des applications d’assistance aux concepteurs dénommés *μ-outils*. Cette plate-forme (PLACID : PLAteforme pour La Conception Innovante et Distribuée) est développée dans le but d’apporter une assistance au travail de co-conception, guidé ou non par des processus complexes de type workflow pour leur capacité à gérer des flots de travaux coopératifs (contrôle et exécution de procédés coopératifs). L’utilisation du paradigme agent concerne aussi bien la modélisation et le développement des différentes couches de la plate-forme que celles des interfaces. A ces objectifs s’ajoutent des contraintes fortes de souplesse et d’adaptabilité, visant à faciliter l’intégration de nouveaux outils de travail collaboratif.

Mots Clés— systèmes multi-agents, agents communicants, agents coopérants, plate-forme collaborative, co-conception distribuée, *μ-outils*.

1 INTRODUCTION

De façon générale, le travail coopératif et les activités collaboratives intégrées dans des salles virtuelles nécessitent des outils de :

- communication interpersonnelle ou de groupe (de types synchrones et/ou asynchrones),
- organisation et cohésion (*ie*, de coordination) des groupes et des activités,
- distribution et partage d’informations, d’applications et de ressources,
- définition spatio-temporelle de la coopération : distance spatiale entre les membres d’une équipe (réalité géographique ou virtualité d’une salle), et distance temporelle dans l’interaction (séquentialité et/ou parallélisme de la réalisation des tâches).

Nous relatons dans cet article la conception d’une plate-forme adaptée à la conception innovante et distribuée, supportant des applications d’assistance aux concepteurs dénommés micro-outils (*μ-outils*) [15]. Le concept de *μ-outils* consiste en des applications logicielles légères, faciles d’utilisation, insérables dans un environnement partagé, connectées entre elles à l’aide d’une base de données. La plate-forme (PLACID : PLAteforme pour La Conception Innovante et Distribuée) est développée dans le but d’apporter une

assistance au travail de co-conception, guidé ou non par des processus complexes de type workflow pour leur capacité à gérer des flots de travaux coopératifs (contrôle et exécution de procédés coopératifs). A ces objectifs s’ajoutent des contraintes fortes de souplesse et d’adaptabilité, visant à faciliter l’intégration de nouveaux outils de travail collaboratif. Le système s’appuie sur une approche orientée agent de la distribution des composants.

Les principales caractéristiques des agents (autonomie, adaptabilité, coopération et communication) permettent, d’une part, de gérer efficacement des composants distribués, hétérogènes et autonomes, et, d’autre part, de faciliter les échanges d’informations et le partage de ressources entre les composants (interaction, communication et coopération). Les agents sont de type *application* (gestion des *μ-outils* et autres outils d’aide à la co-conception), *coordinateur/médiateur*, *système* et *interface*. Le système d’agents doit, quant à lui, assurer l’organisation et le contrôle de la communauté d’agents. L’utilisation effective du système (via une interface elle-même conçue selon une approche agent) se fera dans un contexte d’interactions fortes et

multiples, de multi-utilisateurs et de multi-modalité.

Les μ -outils supportés par la plateforme ne seront pas nécessairement intégrés dans un processus prédéfini de conception. Leur utilisation pourra être simplement ponctuelle, apportant un service bien ciblé dans une phase de conception. Quoiqu'il en soit, chaque μ -outil (conçu dans notre laboratoire) sera connecté au système multi-agent par l'intermédiaire d'un agent hôte. Celui-ci servira d'interface de communication (entrées/sorties) entre les μ -outils et le système d'information coopératif.

Cet article est structuré comme suit : dans la section 2 nous présentons les différents concepts impliqués dans le travail collaboratif assisté par ordinateur (TCAO). La section suivante est consacrée à la description des capacités à communiquer et à interagir des agents, au travers notamment d'un modèle de communication. La section 4 présente les objectifs de la plate-forme PLACID, la spécification des interfaces et la conception du premier μ -outil qu'elle supporte : le « papoticiel ». Finalement, en section 5, nous évoquons les prolongations de notre travail.

2 LE TRAVAIL COLLABORATIF

2.1 Les concepts du travail collaboratif

Le développement des technologies informatiques, la démocratisation de l'Internet, l'utilisation des nouvelles ressources sur Internet ont donné naissance à de nouvelles méthodes de travail. Nous faisons bien entendu, allusion au travail collaboratif ou coopératif assisté par ordinateur (CSCW) [3].

Par définition le collecticiel est un logiciel qui assiste un groupe d'utilisateurs à la réalisation d'un projet commun. Les membres du groupe collaborent à distance, soit au même moment (activité synchrone), soit à des moments différents (activité asynchrone). Les champs d'application sont très nombreux. Ainsi des activités aussi diverses que la conception de produits industriels, l'enseignement, les relations commerciales ou les jeux sont susceptibles d'être assistées par l'ordinateur. Les systèmes informatiques destinés au support du travail coopératif doivent permettre aux utilisateurs de réaliser une tâche collaborative à partir de postes de travail respectifs ou depuis des installations spécialisées (salles de vidéoconférence...).

Les collecticiels doivent permettre à plusieurs utilisateurs de collaborer dans des espaces partagés explicites. Les concepts à considérer sont ainsi :

— Le temps et l'espace : on peut dans un collecticiel vouloir réunir plusieurs personnes distantes géographiquement (bureau à proximité ou distant) ou ne travaillant pas en même temps (rythmes différents, incompatibilité des emplois du temps,...).

— Un ensemble de modes de coopération : la *coopération asynchrone* lorsque les différents acteurs interagissent dans un projet en échangeant des données

et en travaillant quand ils peuvent (mode de travail autonome) ; la *coopération en session* lorsque les différents participants travaillent en même temps sur des données qui leur sont propres, tout en restant accessibles pour communiquer (l'objectif étant de faire diminuer les délais d'interaction entre les différents acteurs d'un projet) ; la *coopération en réunion* lorsque tous les utilisateurs, clairement identifiés, travaillent et communiquent en co-temporalité tout en partageant les objets de leurs travaux et discussions (des rôles leur sont attribués, et chacun participe à son tour) ; la *coopération étroite* lorsque les acteurs communiquent, interagissent en temps réel sur tous les objets partagés du projet (accroissement de la coproduction).

— La flexibilité opérant dans des domaines hétérogènes : dans l'interaction, dans la distribution de données, dans le partage, dans le contrôle d'accès, dans la représentation de l'information, de la planification dans l'exécution des tâches, etc.

2.2 Les fonctionnalités du travail collaboratif

Les activités du travail collaboratif sont les échanges, le partage et la coopération entre participants. Il est d'usage de présenter les systèmes d'information coopératifs comme pouvant répondre aux enjeux de la coopération (ci-dessous 6 enjeux, les 3 premiers pour des coopérations ponctuelles, les 3 autres lorsqu'un groupe se structure et devient une organisation) :

- Faciliter le partage de ressources.
- Assister la coordination.
- Améliorer la communication de groupe.
- Favoriser l'implication individuelle.
- Entretenir la cohésion des équipes.
- Favoriser le développement de l'organisation

La figure suivante (Fig. 1) schématise les liens unissant les fonctionnalités de base d'un collecticiel : la collaboration (visioconférence, vidéoconférence, outils de localisation, d'annotation interpersonnelles, réunion électronique et décision de groupe, édition conjointe), la mémoire de groupe (base d'informations partagées dans un groupe, bases de documents), la circulation de documents et le workflow.

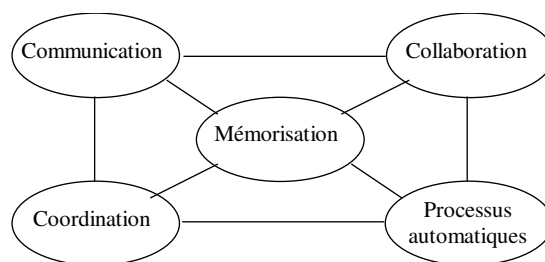


FIG. 1 – Fonctions basiques pour un collecticiel

La nécessité de standards qui permettraient l'interopérabilité d'applications de workflow a depuis un certain temps été reconnue. Dans le contexte de

l'internet, des groupes de travail s'efforcent de proposer de tels standards. C'est autour de la WfMC (*Workflow Management Coalition*) que s'élaborent de telles spécifications empruntant aux approches objets [16].

3 DES AGENTS POUR LA COOPERATION

Le principal intérêt des SMA est qu'ils permettent de distribuer des agents, entités communicantes, autonomes, réactives et dotées de compétences. Pour réaliser un SMA selon ces critères, il faut doter chaque agent, dit « cognitif », des trois propriétés suivantes : indépendance, communication et intelligence (expertise, savoirs-faire). Il nous faut aussi définir l'architecture des agents (fonctions et interactions cognitives), ainsi que la structuration des connaissances nécessaires pour leurs différentes activités.

3.1 Eléments de Modélisation

La définition de nos agents est adaptée du modèle à 3 niveaux de l'opérateur de Rasmussen (comportement réflexe, comportement à base de règles, comportement à base de connaissances avec interprétation, décision et plan). Nous l'avons interprété comme modèle de processus de nos agents dont les comportements sont adaptés aux tâches qu'ils réalisent :

— **Def Agent** ::= <Communication/langage, Perception, Buts/intentions, Décision/plan, Mémoire, Identification/Interprétation, Actions/Réactions>.

3.2 Architecture d'un Agent Cognitif

Dans [6] nous avons proposé l'architecture générale d'un agent cognitif, respectant les trois propriétés d'indépendance, de communication et d'intelligence. Celle-ci (Fig. 2), inspirée de la théorie de modularité de J. Fodor est composée de cinq modules gérant les connaissances, la perception, la communication, le contrôle et le raisonnement de l'agent, dont nous avons proposé un modèle réseau de Petri.

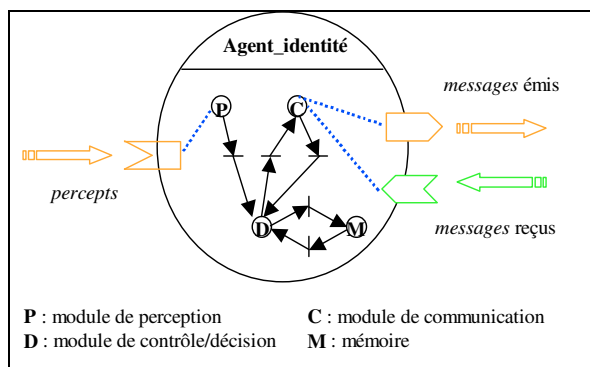


FIG. 2 – Architecture modulaire d'un agent

Nous ajouterons à cette description que les agents sont des entités hétérogènes aux modes d'interactions variés et aux comportements complexes. Une modélisation de SMA doit aussi définir le type

d'organisation des agents et sa capacité d'évolution.

3.3 Des Agents Communicants

La communication est le principal mécanisme d'interactions d'un agent avec la communauté des agents – en conformité avec la relation triangulaire (*Communication, Coopération, Compétition*). Nous insistons sur le fait que pour qualifier un agent d'"intelligent" il est essentiel de prouver ses capacités à communiquer dans un but individuel ou collectif.

Pour communiquer entre eux (information ou dialogue pour la coopération), les agents expriment leurs intentions selon le langage KQML, dérivé de la théorie des actes de langage. La forme générale d'un acte de langage est donnée par J. Searle sous l'expression $F(p)$, avec $F = \{Affirmer, Demander, Promettre, Exprimer, Déclarer\}$ et p une proposition. Le format que nous avons retenu est défini par le quintuplet <intention, émetteur, récepteur, langage, message>. Il permet de représenter le contexte, l'intention et le message de la communication.

Le schéma canonique de communication proposé par Abraham Moles, augmenté du feed-back et de la constitution de schémas de croyances, contient les éléments intervenant dans l'acte de communication : intentionnalité et communication réciproque.

La représentation des connaissances d'un agent élaborée dans un contexte de communication/dialogue, respectant le modèle des actes de langage, est réalisée sous la forme d'un schéma de croyances. Celui-ci est constitué d'un réseau de *frames* avec des niveaux de croyances/connaissances (*{Croire, Savoir, Déduire}*) établis lors de la communication par l'intermédiaire des actes illocutoires définis dans la taxinomie de Searle.

3.4 Des Agents Coopérants

Les systèmes de travail collaboratif sont constitués de composants distribués, hétérogènes et autonomes. Les systèmes développés en intelligence artificielle distribuée (IAD), et notamment les SMA sont donc bien adaptés. Leur apport potentiel des agents concerne :

- la prise en charge d'actions répétitives et la délégation de tâches sans intérêt pour l'utilisateur,
- la prise de décision par compréhension du contexte d'utilisation (pertinence),
- la personnalisation de l'information (préférences, buts et capacités de l'utilisateur),
- l'interactivité plus naturelle (modalités, forme et présentation),
- et l'adéquation aux systèmes en réseau, notamment coopératifs.

Les comportements individuels et coopératifs des agents sont variés : initialisations, planification des actions, émission et réception de documents et de messages, recherche de documents ou d'information, supervision de procédures. Chacun de ces services correspond à la mise en œuvre de compétences.

3.5 Des Agents pour Modéliser les Interfaces

Les modèles multi-agents pour système interactif répondent à des préoccupations communes : modularité, distinction entre présentation et abstraction, encapsulation des objets graphiques, parallélisme.

Un système interactif s'analyse usuellement selon trois facettes complémentaires : le noyau fonctionnel, le contrôleur de dialogue et la présentation. Un agent PAC [4] se modélise selon ces trois perspectives, renommées Présentation, Abstraction et Contrôle (Fig. 3).

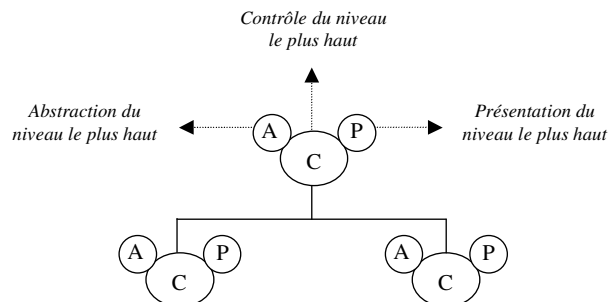


FIG. 3 – Structuration hiérarchique des agents PAC.

PAC [4] fournit un cadre de construction systématique applicable à tous les niveaux d'abstraction d'un système interactif. L'approche récursive permet de concevoir une architecture progressivement de façon ascendante ou descendante. Enfin le modèle PAC permet de traduire l'encapsulation/ affinement (hiérarchie d'agents liée au niveau de détail auquel le concepteur logiciel s'intéresse) et la dépendance fonctionnelle, deux activités de conception logicielle étroitement liées.

3.6 Le Processus d'Agentification

Les agents sont des entités possédant un certain nombre de compétences qui leur permettent de jouer un ou plusieurs rôles dans une organisation. Ils sont regroupés au sein d'un SMA organisé selon une structure hiérarchique (agents *spécialistes*, agents *médiateurs* et agents *superviseurs*).

Pour la spécification du SMA nous reprenons les propositions faites dans la définition du langage A_UML [11], ainsi que nos propres propositions méthodologiques [6] (schématisées en Fig. 4) :

- 1) réaliser le diagramme de cas d'utilisation (services rendus par le système), et pour chacun des usages identifiés réalisés les 3 phases suivantes ;
- 2) réaliser le diagramme de classes mettant en relation les agents concernés par l'usage (on peut aussi faire usage du diagramme de collaboration) ;
- 3) définir le comportement de chaque agent au moyen d'un diagramme d'états ou d'activités ;
- 4) sur la base de scénarios d'usage, réaliser les diagrammes de séquence qui précisent les échanges de messages (et leur ordonnancement) entre les agents concernés par les scénarios.

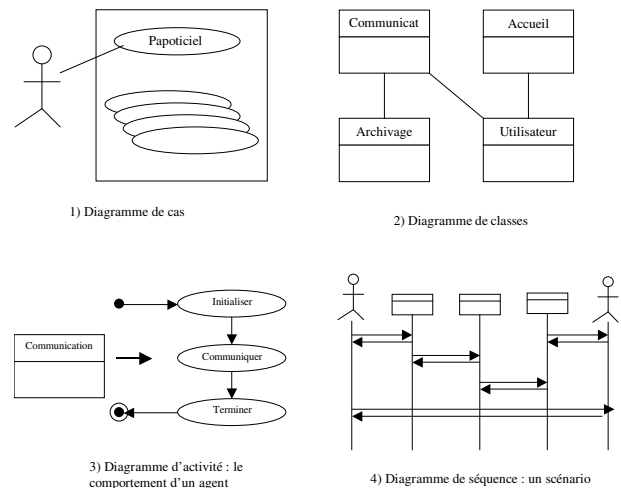


FIG. 4 – Méthodologie suivie pour l'exemple du « Papoticiel » décrit plus loin (cf. §4.4.).

4 LA PLATEFORME PLACID

4.1 Objectifs du Projet

Le projet PLACID (Plate-forme Logicielle d'Aide à la Conception Innovante et Distribuée) consiste en la définition et le développement d'une plate-forme logicielle apportant un certain nombre de services pour l'utilisation d'un environnement de co-conception virtuelle (partage d'objets, services de gestion de tâches, services de communications et peut-être outils d'aide à la décision). Le schéma préliminaire de PLACID est présenté ci-dessous (Fig. 5).

L'architecture de PLACID est composée de 4 couches, offrant par sa modularité des possibilités d'utilisation multi-plateformes et d'évolutivité :

- Couche 1 : c'est la couche de présentation du système de conception dans un contexte d'interactions multiples, multi-utilisateurs et multi-modal.
- Couche 2 : couche de gestion des outils d'aide à la co-conception (μ -outils et autres outils).
- Couche 3 : couche de gestion des travaux collaboratifs (de type workflow), permettant de contrôler et exécuter les procédés coopératifs.
- Couche 4 : couche du système d'exploitation et de gestion des communications de bas niveau.

Pour la gestion des données nous avons adopté une architecture adaptée aux SIC (Systèmes d'Information Coopératifs), à savoir : un SGBD fédérant des multi-bases. Ceci permet de concevoir efficacement des applications globales ou locales.

4.2 Description de la Plateforme PLACID

La définition du projet PLACID répond à la démarche "équipe projet" instituée dans le laboratoire et s'intègre dans les 2 axes (points de vue sur le produit et coopération en conception) de notre équipe afin de faciliter l'utilisation de μ -outils de conception par une équipe de concepteurs proches ou distants, dans le

cadre de la conception distribuée structurée en modules (fonctionnel, structurel, fabrication, maintenance). L'ensemble constituant un véritable "bureau de co-conception virtuelle".

La figure 6 présente l'architecture agent de la première couche de la plate-forme reliée à un ORB

chargé de la gestion des échanges et du partage des informations. Cette couche d'agents se décompose en deux niveaux : des agents *acteurs*, proches des utilisateurs et des μ -outils, et des agents *spécialistes* (exécutants) dotés des compétences indispensables à la coopération.

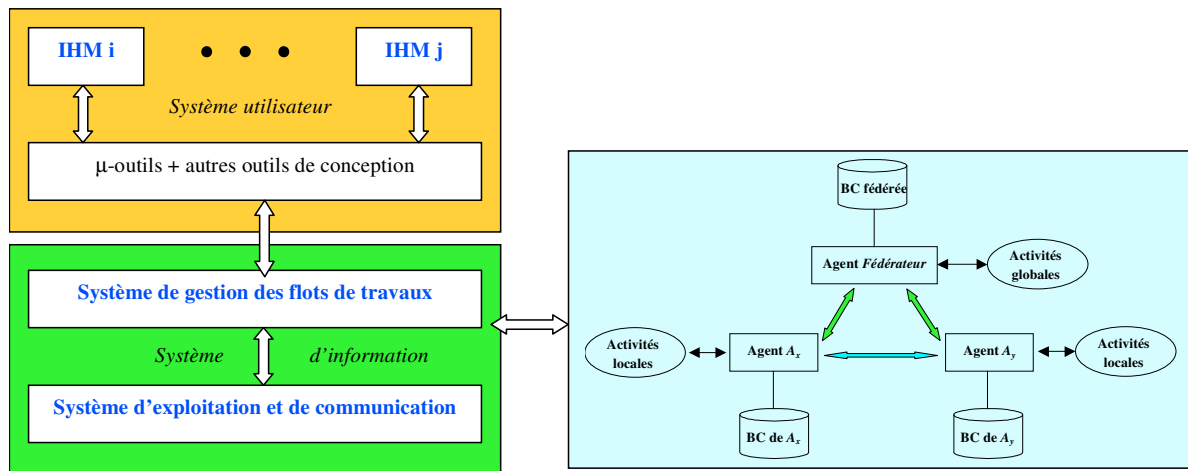


FIG. 5 – Architecture de principe de la plate-forme

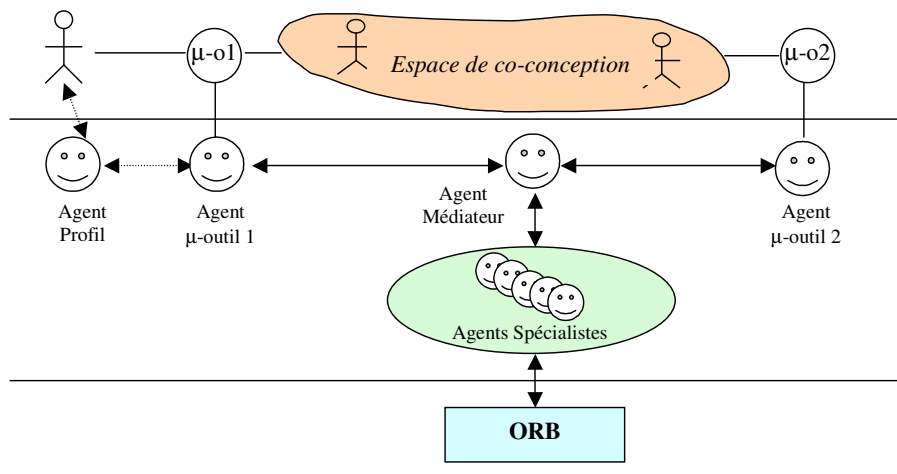


Fig. 6 – Architecture agent du système d'information coopératif de PLACID

4.3 Spécification des Interfaces

4.3.1 Eléments méthodologiques

Pour le cycle de développement des interfaces des μ -outils nous avons retenu la démarche incrémentale : développement successif avec amélioration de prototypes. Cette démarche permet d'évaluer lors de chaque phase terminale d'un incrément la pertinence de l'interface et le respect des objectifs qui ont conduit l'incrément. L'analyse des tâches utilisateurs rentrant dans la spécification finale de chaque μ -outils nous adoptons la méthode SADT/Petri [1]. En effet, cette association a été conçue pour réaliser l'analyser, la

spécification et l'évaluation des systèmes interactifs comme les IHM. L'intérêt majeur de cette association est d'effectuer la structuration fonctionnelle du système en termes de tâches par SADT, et de décrire son comportement dynamique par les réseaux de Petri.

Une fois l'IHM spécifiée il s'agit de concevoir son architecture logicielle. Nous adoptons pour celle-ci le modèle PAC [4], particulièrement adapté à la modélisation des architectures logicielles des systèmes interactifs et/ou multimodaux et des collecticiels.

4.3.2 Interface de PLACID

Dans la figure suivante (Fig. 7), nous avons

volontairement dissocié la description de l'interface de la plate-forme PLACID (environnement/bureau de travail collaboratif), des interfaces des différents μ -outils qui seront supportés par la plate-forme. De même nous ne considérerons que les besoins associés à

l'accès simplifié et à la présentation claire et homogène des applications supportées par la plate-forme et des outils de collaborations définis pour la première version de PLACID.

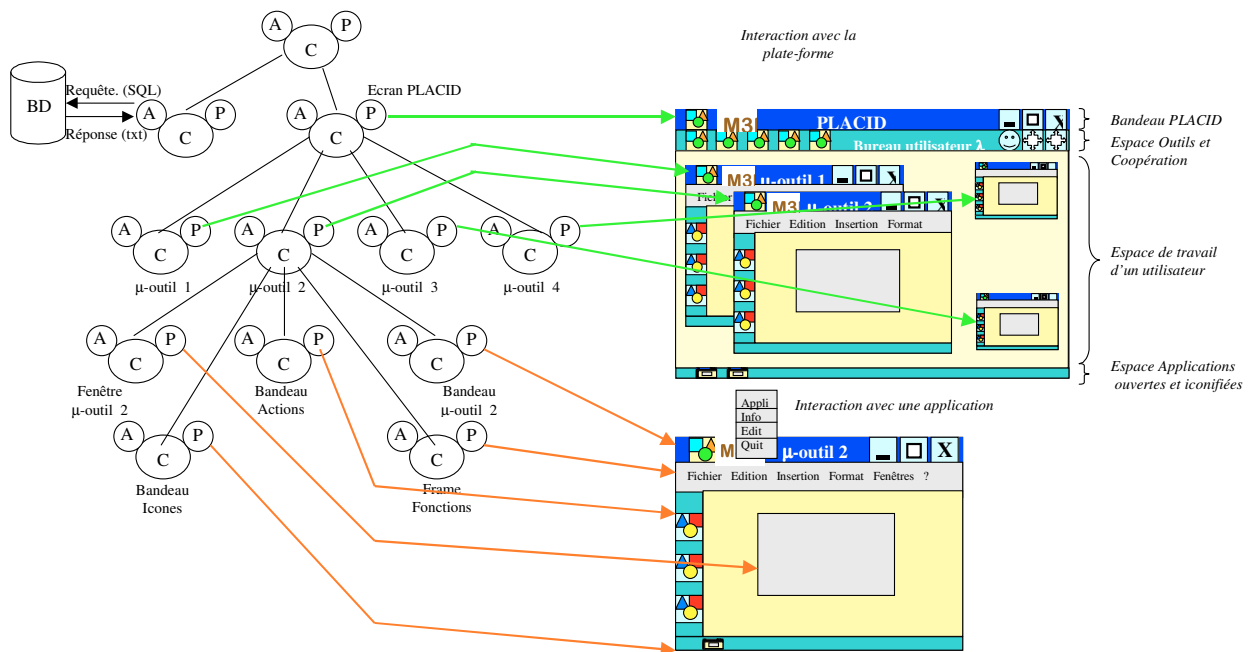


FIG. 7 – Architecture PAC pour la spécification des interfaces des μ -outils supportés par PLACID

4.4 Un Exemple de Micro-Outil : le Papoticiel

Pour concevoir et tester le premier prototype de PLACID nous avons choisi de modéliser une application de type « Papoticiel ». En effet cette application de réunion électronique, outre son aspect communicationnel, met en relation des applications de coopération aussi variées que la gestion d'un groupe d'utilisateurs, la maintenance d'un agenda, la gestion d'une mémoire de groupe au travers de l'archivage des réunions et de ces éléments, ainsi que l'édition partagée de compte-rendu de réunion. Nous présentons donc ci-dessous, quelques éléments de conceptions de notre « papoticiel » :

- le diagramme de cas d'utilisation (Fig. 8.a) pour fixer le contexte d'utilisation du *papoticiel*. L'application *papoticiel* peut être déclenchée à l'initiative d'un membre du groupe, qualifié d'*initiateur*, ou par l'agent logiciel *agenda*.

- le datagramme SADT (Fig. 9) pour représenter le modèle de tâches d'utilisation du *papoticiel*,

- le diagramme de classes du *papoticiel* (Fig. 10) pour définir l'architecture agent et les inter-relations entre agents (celui-ci peut être complété par un diagramme de collaboration entre agents),

- et enfin le diagramme de séquence *Se réunir* (scénario « init-fin ») déclenché par un membre *initiateur* (Fig. 8.b), servant à illustrer un des scénarios possibles d'utilisation du *papoticiel*.

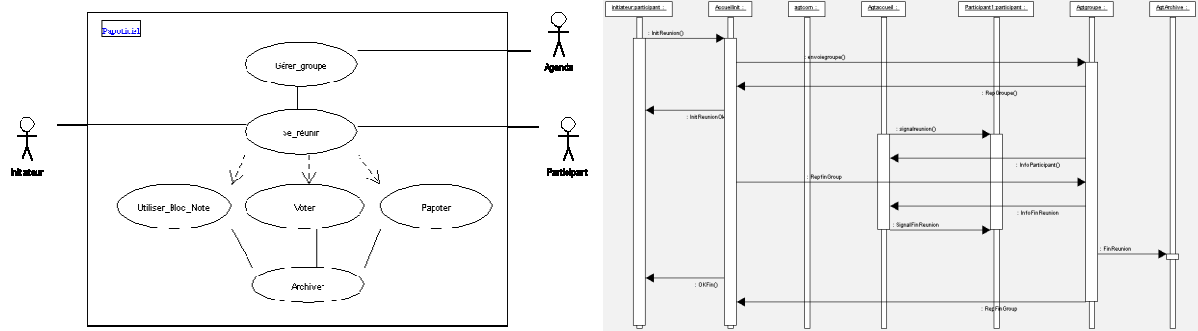


FIG. 8 – a) Diagramme de Use Cases du *Papoticiel* b) Diagramme de séquence *Se_réunir*.

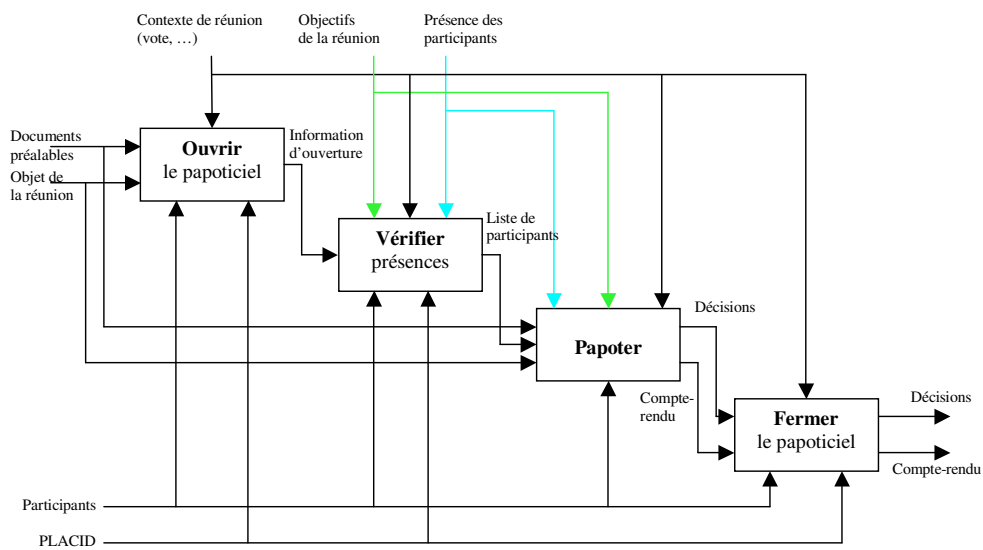


FIG. 9 – Décomposition SADT de l'analyse de notre tâche test «Utiliser le papoticiel ».

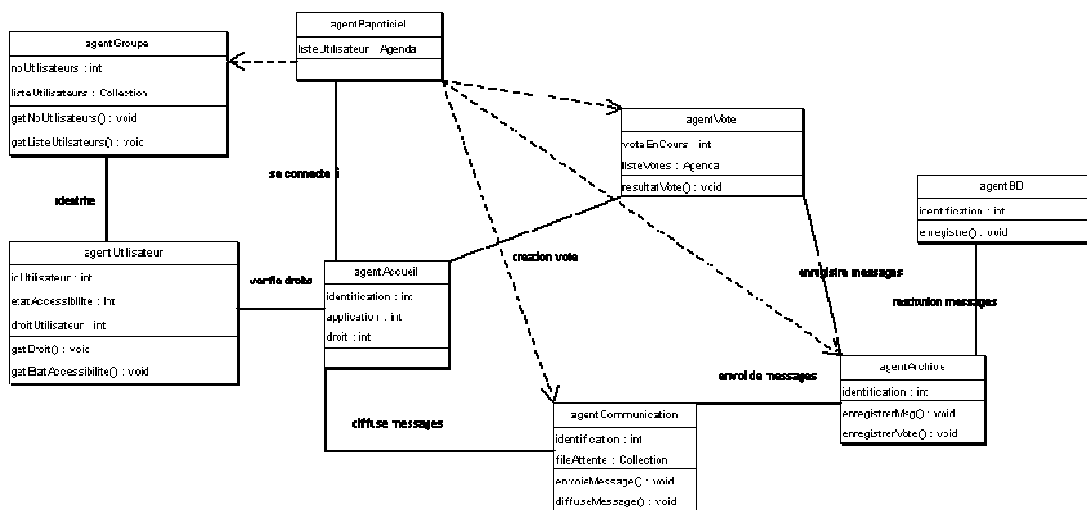


FIG. 10 – Diagramme de classes du *Papoticiel*, selon notre architecture agent (cf. figure 6)

Remarque : pour l'application *Papoticiel*, illustrée par les figures précédentes et notamment le diagramme de classes, nous avons fait appel à 8 types d'agents :

- 1 agent application : l'agent papoticiel,
- 2 agents utilisateur pour gérer plus spécifiquement la séance de travail des 2 utilisateurs (initiateur et participant) ;
- 2 agents d'accueil pour gérer et accompagner les arrivées et départs des membres d'un groupe d'utilisateurs ;
- 1 agent groupe : pour gérer et coordonner l'activité de groupe (ici réunion),
- 1 agent communication pour gérer le flux des messages échangés par l'ensemble des agents (envoi, réception, accusé de réception, ...) ;
- 1 agent pour les votes : pour gérer les consultations nécessaires à une prise de décision collective,
- 1 agent d'archivage pour archiver les messages selon 2 critères : les messages fonctionnels pour la maintenance de la plateforme et les messages entre utilisateur pour rédiger (dans un premier, aider par la suite) le compte-rendu de la réunion électronique ;
- 1 agent base de données pour gérer les échanges (requêtes/réponses) avec la base.

4 CONCLUSION

Du bilan d'une étude menée sur les perspectives du travail collaboratif et des technologies agents est ressorti un ensemble de concepts prometteurs qui ont servi à la définition des objectifs de PLACID :

- Concernant le TCAO : définition d'un projet de co-conception distribué offrant des services de partage de ressources, coordination de groupe, réunion électronique, décision de groupe, compte-rendus et mémoire de groupe (ou de conception).
- Concernant les SMA : distribution d'activités coopératives et de composants d'aide à la conception ; aide à la décision dans le processus de conception distribué et prise en charge d'actions (tâches répétitives ou implicites dans un cadre de coopération), conception des IHM.

A la suite de cette étude et des premiers éléments de modélisation de PLACID, nous avons réalisé un prototype de la plate-forme qui permet d'utiliser et de coordonner un « Papoticiel » élémentaire et un outil d'assistance à la « gestion de groupes » faisant appel aux services d'une base de données. Ainsi une « réunion électronique artificielle » sert de scénario de tests à notre prototype, en attendant que l'ensemble des µ-outils existants et en développements ne soient progressivement intégrés à la plate-forme.

REFERENCES

- [1] ABED M., EZZEDINE H., KOLSKI C., *Modélisation des tâches dans la conception et l'évaluation des systèmes interactifs :*

- la méthode SADT/Petri*, in *Analyse et conception de l'IHM*, sous la direction de Christophe Kolski, Hermès, 2001.
- [2] CANALDA P., CHATONNAY P., FOUGERES A.-J., *Pédagogie de projets tutorés basée sur la synchronisation de fragments de procédés coopératifs : motivation, modélisation et expérimentation*, Workshop ARIADNE, Lyon, 13-15 novembre 2002.
- [3] COURBON J.-C., TAJAN S., *Groupware et intranet. Vers le partage des connaissances*, 2^{ème} édition, DUNOD, 1999.
- [4] COUTAZ J., NIGAY L., *Architecture logicielle conceptuelle des systèmes interactifs*, in *Analyse et conception de l'IHM*, sous la direction de Christophe Kolski, Hermès, 2001.
- [5] FOUGERES A.-J., *Model of cognitive agents to simulate complex information systems*, IEEE International Conference on Systems, Man and Cybernetics, (SMC'02), Hammamet, Tunisia, October 6-9 2002.
- [6] FOUGERES A.-J., *Des agents communicants pour simuler et détecter des épidémies*, Revue ISI, (8)1, Hermès, 2003.
- [7] GRISLIN-LE STRUGEON E., ADAM E., KOLSKI C., *Agents intelligents en interaction homme-machine dans les systèmes d'information*, in *Environnements évolués et évaluation de l'IHM*, sous la direction de Christophe Kolski, Hermès, 2001.
- [8] HERIN D., ESPINASSE B., ANDONOFF E. et HANACHI C., *Des systèmes d'information coopératifs aux agents informationnels*, in *Ingénierie des systèmes d'information*, sous la direction de Corine Cauvet et Camille Rosenthal-Sabroux, Hermès, 2001.
- [9] HOOGSTOEL F., *Une approche organisationnelle du travail coopératif assisté par ordinateur. Application au projet CO-LEARN*, Thèse de Doctorat de l'Université des Sciences et Techniques de Lille, 1995.
- [10] NIGAY L., *Conception et modélisation logicielle des systèmes interactifs : application aux interfaces multimodales*, Thèse de Doctorat de l'Université de Grenoble 1, IMAG, 1994.
- [11] ODELL J., PARUNAK H.V.D., BAUER B., *Extending UML for agents*, Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence, Austin, Texas, July, 30, 2000.
- [12] PALANQUE P. et BASTIDE R., *Spécifications formelles pour l'ingénierie des interfaces homme-machine*, RSR-CP, 13/2001, 207-232, Hermès, 2001.
- [13] ROUTIER J.-C. et MATHIEU P., *Une contribution du multi-agent aux applications de travail coopératif*, Technique et science informatiques, 14(4), 473-500, 1995.
- [14] Rapport d'Activité, *Projet ECOO. Environnements et COOpération*, INRIA Lorraine, Nancy, 2001.
- [15] VAN HANDENHOVEN E., TRASSAERT P., *Design knowledge and design skills*, International Conference on Engineering Design (ICED 99), Munich, Allemagne, 24-26 août, 1999.
- [16] WPMC, *Workflow Management Coalition, Terminology & Glossary*, WPMC-TC-1011, Issue 3.0., 1999.